Video-Database of Emergents

Exploring Emergence: Video-Database of Emergents Observed in Advanced Cellular Automaton 'Game of Life' Using GoL-N24 Software

Author: Jiří Kroc

Independent Researcher, Complex Systems Research
Pilsen, The Czech Republic.

email: dr.j.kroc@gmail.com
www: https://www.researchgate.net/profile/Jiri-Kroc

Created: September 21, 2023
Updated: February 28, 2024

Abstract

The video-database of animations of emergents produced by the open-source, Python GoL-N24 and error-resilient r-GoL programs that are simulating extended and generalized versions of John. H. Conway's 'Game of Life'. The video-database provides links to all important animations and figures presented in current publications and to future ones based on the r-GoL and GoL-N24 software [9, 8]. The video-database contains even unpublished animations. This video-database aims to provide the concise, easy-to-navigate way to find emergents observed in this cellular automaton. The related database of initial files for animations of emergents can be found in [17].

Cite as: J. Kroc: Emergent Information Processing: Observations, Experiments, and Future Directions, TBA, (2024).

link: https://www.researchgate.net/publication/373044641

Contents

List of Tables

Beware: This video-database of animations observed within GoL-N24 software is accompanied by the database of the initial files [17] that can be simply loaded into the GoL-N24 software. Both databases—this one (videos) and the one with the initial configurations [17]—are going to be updated in the future along with research progress.

Chapter 1

Introduction into the Cellular Automata: GoL, r-GoL, and GoL-N24

1.1 The Structure of the Database

The whole database is introduced by links to the introductory cellular automata theory books and articles. This is followed by the 'The simplest Python Program Simulating GoL' (see [7, 30, 32]) and by the error-resilient variants of GoL (see r-GoL program [9]) to enable everyone to dive quickly and easily into this type of programming. Experts and those familiar with cellular automata programming can directly continue in Sections 1.4 and 1.5.

1.2 Brief Introduction into Cellular Automata

A simple introduction that addresses the modeling of complex systems can be found in the book [31], where all major methods used to model complex systems are covered in a highly accessible way.

A brief and concise introduction to cellular automata can be found in [29], which is recommended to be followed by reading the following articles [33, 34] and the book [2]. A very rudimentary definition of a cellular automaton is: "A cellular automaton is made of a matrix or lattice of elements that are

updating their current state according to the states of elements within a uniform neighborhood and the element itself using an identical evolution function. This is all happening in discrete time steps."

1.3 The Simplest Open-Source, Python 'GoL' Program

For educational reasons, the simplest open-source Python 'GoL' program [7], which is less than one hundred lines long, was developed to explain the very principles of cellular automata programming in John. H. Conway's 'Game of Life' [1] famous example.

Those who are already familiar with this type of programming can skip this section and go directly to Section 1.4, which describes an avenue towards the development of error-resilient computations.

All available variants of open-source, Python GoL software

Soft Type	**Software – Brief Description**	**Links**
GoL	The simplest GoL software (below 100 lines of the code).	software and manual [7]
GoL-N24	Interactive GoL software GoL-N24 with extended neighborhoods selected from (5 x 5) neighbors.	software [8]
r-GoL	Error-resilient r-GoL software, which demonstrates resilience against 1% of injected errors, is accomplished by modification of the updating rule.	software [9]

Table S1: To enable easy navigation, links to the full, open-source Python software GoL-N24 [8], the simplest GoL program [7], and error-resilient software [9] are provided in this table.

1.4 Error-Resilient 'GoL' Software Variants (r-GoL)

The following animations and figures were generated using the software [9] that were published in [11]. Robust emergent r-GoL figures can be found here [10]; see Table S2.

Logic-Gate Type	**Cellular Automaton – Brief Description**	**Citation**
r-GoL	Animations of r-GoL, rule-A without injected errors, Figures 3 and 4 in [11] and the animation link in [4].	[10]
r-GoL	Animations of r-GoL, rule-B with 1% of injected errors, Figures 5 and 6 in [11] and the animation link in [5].	[10]
r-GoL	Animations of r-GoL, rule-C with 1% of injected errors, Figures 7 and 8 in [11] and the animation link in[6].	[10]

Table S2: Three basic simulations demonstrate the error-resilient variants of 'Game of Life'. The rules B and C are error-resilient against the injection of one percent of error into the updating rule; see details in [11]!

1.5 Open-Source, Python 'GoL-N24' Program

The full, open-source Python software GoL-N24 [8] was developed in the interactive version. It enables loading and saving configurations of the lattice and the definition of the neighborhood, starting/stopping the simulation, editing neighborhood, and randomly generating the initial configuration of the lattice.

Just a few examples of the emergents observed within the extended neighborhood of GoL-N24:

- Ships breeding ships [12] is a prominent example of a second-order emergents generated by spontaneously arising first-order emergents from almost any random initial conditions.

- Manually defined, there are three emergent logic gates implemented in the original GoL: AND [13], OR [15], and NOT [14].
- Resilience of a glider against small changes within the neighborhood [28].
- and the list goes on...

Chapter 2

Published Research on Emergent Information Processing

This database is drawing animations from the following publications; for details, see Table S3.

Cellular Automaton – Brief Description	**Citation**
Emergent Information Processing: Observations, Experiments, and Future Directions	[16]
Robust massive parallel information processing environments in biology and medicine: case study	[11]
Complex Systems and Their Use in Medicine: Concepts, Methods and Bio-Medical Applications	[3]

Table S3: The list of publications on Emergent Information Processing. An important review of applications of complex systems modeling in medicine, which provides a good starting point for understanding complex systems modeling in biology.

Chapter 3

The Video-Database of Emergents Observed in GoL-N24

This video-database is going to be extended, along with finding more interesting emergents observed in the GoL-N24 cellular automaton.

Important message: Animations are located at the bottom of the pages on 'ResearchGate.net' in APNG formats, which can be displayed by web browsers from the local files.

3.1 Logic-gates AND, OR, and NOT implemented in GoL

The animations of the logic gates AND S4, OR S5, and NOT S6 are going to be reviewed in this section. This will enable everyone to easily and quickly find all required animations for any logic-gate input.

It is important to mention that there are existing projects attempting to design a functional processor and memory fully defined by logic gates in a CA-lattice of the original GoL; see the publication [16] for details and links. Those attempts are supported by the proof that GoL is the Turing machine equivalent; see the same reference [16].

Logic-Gate Type	Cellular Automaton – Brief Description	Citation
AND	logic gate AND – input 00, animation link [18]	[13]
AND	logic gate AND – input 01, animation link [19]	[13]
AND	logic gate AND – input 10, animation link [20]	[13]
AND	logic gate AND – input 11, animation link [21]	[13]

Table S4: Logic gate AND was implemented in the original GoL using the software GoL-N24 [8] (alternatively, the simplest GoL program [7] can be used). Animations are located at the bottom of the pages in APNG formats, which can be displayed by web browsers.

Logic-Gate Type	Cellular Automaton – Brief Description	Citation
OR	logic gate OR – input 00, animation link [24],	[15]
OR	logic gate OR – input 01, animation link [25],	[15]
OR	logic gate OR – input 10, animation link [26],	[15]
OR	logic gate OR – input 11, animation link [27],	[15]

Table S5: Logic gate OR was implemented in the original GoL using the software GoL-N24 [8] (alternatively, the simplest GoL program [7] can be used). Animations are located at the bottom of the pages in APNG formats, which can be displayed by web browsers.

3.2 Overview of Important and Interesting Emergents Found in GoL-N24

This is the main section of this text. Gradually, as the research goes on, newly discovered emergents will be presented here in Table S7. Everyone

Logic-Gate Type	**Cellular Automaton – Brief Description**	**Citation**
NOT	logic-gate NOT – input 0, animation link [22]	[14]
NOT	logic-gate NOT – input 0, animation link [23]	[14]

Table S6: Logic gate NOT was implemented in the original GoL using the software GoL-N24 [8] (alternatively, the simplest GoL program [7] can be used). Animations are located at the bottom of the pages in APNG formats, which can be displayed by web browsers.

who finds some interesting emergents is welcome to contact the database author and send the relevant data. It will be included with all credits and links.

Gate Type	**Cellular Automaton – Brief Description**	**Citation**
2^{nd} order emergents	Ships breeding ships: the first instance of the observed 2^{nd}-order emergents within GoL-N24	[12]
neigh. resilience	Robust emergents within GoL resisting perturbation of neighborhood	[28]

Table S7: Interesting emergents were found within the extended neighborhood of GoL using the software GoL-N24 [8] and error-resilient r-GoL [9].

Bibliography

[1] M. Gardner. The fantastic combinations of John Conway's new solitaire game "life". *Scientific American Magazine*, 223(10):120–123, 1970. https://www.scientificamerican.com/article/mathematical-games-1970-10/.

[2] A. Illachinski. *Cellular Automata: A Discrete Universe*. World Scientific, 2001. https://www.worldscientific.com/worldscibooks/10.1142/4702#.

[3] J. Kroc. Complex Systems and Their Use in Medicine: Concepts, Methods and Bio-Medical Applications. preprint on ResearchGate, www.researchgate.net/publication/330546521, Jan 2019.

[4] J. Kroc. Animations of r-GoL, rule-A without ejected errors, Figures 3 and 4. https://www.researchgate.net/publication/357285926, 202. Open file GoL-anim-dif-v1-1-stabil-tst1-fin-Fig-3-4.png, ResearchGate, Accessed as on 09-15-2023.

[5] J. Kroc. Animations of r-GoL, rule-B with 1% of ejected errors, Figures 5 and 6. https://www.researchgate.net/publication/357285926, 202. Open file GoL-anim-dif-v1-1-stabil-tst1-fin-Fig-5-6.png, ResearchGate, Accessed as on 09-15-2023.

[6] J. Kroc. Animations of r-GoL, rule-C with 1% of ejected errors, Figures 7 and 8. https://www.researchgate.net/publication/357285926, 202. Open file GoL-anim-dif-v1-1-stabil-tst1-fin-Fig-7-8.png, ResearchGate, Accessed as on 09-15-2023.

[7] J. Kroc. The simplest Python program simulating a cellular automaton model of a complex system: the 'Game of Life'. https:

//www.researchgate.net/publication/355043921, Oct 2021. Accessed as on 03-25-2023.

[8] J. Kroc. Exploring Emergence: Python program GoL-N24 simulating the 'Game of Life' using 8 Neighbors from 24 Possible. https://www.researchgate.net/publication/365477118, Nov 2022. Accessed as on 03-25-2023.

[9] J. Kroc. Python program simulating cellular automaton r-GoL that represents robust generalization of 'Game of Life'. https://www.researchgate.net/publication/358445347, Feb 2022. Accessed as on 03-25-2023.

[10] J. Kroc. Python program simulating cellular automaton r-GoL that represents robust generalization of 'Game of Life': sample runs. https://www.researchgate.net/publication/357285926, Feb 2022. Accessed as on 03-25-2023.

[11] J. Kroc. Robust massive parallel information processing environments in biology and medicine: case study. *Journal of Problems of Information Society*, 13(2):12–22, 7 2022. https://doi.org/10.25045/jpis.v13.i2.02.

[12] J. Kroc. Emergent Computations: Emergents Are Breeding Emergents as Demonstrated on Ships Breding Trains of Ships Occuring in Modified GoL Using Program GoL-N24. https://www.researchgate.net/publication/368635079/, Feb 2023. ResearchGate, Accessed as on 04-01-2023.

[13] J. Kroc. Emergent computations: simulations of logic-gate AND using cellular automaton GoL-N24 implemented in Python. https://www.researchgate.net/publication/368300518/, Feb 2023. ResearchGate, Accessed as on 03-25-2023.

[14] J. Kroc. Emergent computations: simulations of logic-gate NOT using cellular automaton GoL-N24 implemented in Python. https://www.researchgate.net/publication/368300097/, Feb 2023. ResearchGate, Accessed as on 03-25-2023.

[15] J. Kroc. Emergent computations: simulations of logic-gate OR using cellular automaton GoL-N24 implemented in Python. https:

//www.researchgate.net/publication/367380336/, Jan 2023. ResearchGate, Accessed as on 043-01-2023.

[16] J. Kroc. Emergent Information Processing: Observations, Experiments, and Future Directions. preprint on ResearchGate, www.researchgate.net/publication/373044641, Aug 2023.

[17] J. Kroc. Exploring Emergence: Database of Initial Files of Advanced Cellular Automaton 'Game of Life' Using GoL-N24 Software. https://www.researchgate.net/publication/372187009, Sep 2023. Accessed as on 09-10-2023.

[18] J. Kroc. Logic-gate AND: inputs 00. https://www.researchgate.net/publication/368300518/, Sep 2023. Open file GoL-AND-00-anim.png, ResearchGate, Accessed as on 09-15-2023.

[19] J. Kroc. Logic-gate AND: inputs 01. https://www.researchgate.net/publication/368300518/, Sep 2023. Open file GoL-AND-01-anim.png, Accessed as on 09-15-2023.

[20] J. Kroc. Logic-gate AND: inputs 10. https://www.researchgate.net/publication/368300518/, Sep 2023. Open file GoL-AND-10-anim.png, Accessed as on 09-15-2023.

[21] J. Kroc. Logic-gate AND: inputs 11. https://www.researchgate.net/publication/368300518/, Sep 2023. Open file GoL-AND-11-anim.png, ResearchGate, Accessed as on 09-15-2023.

[22] J. Kroc. Logic-gate NOT: inputs 00. https://www.researchgate.net/publication/368300097/, Sep 2023. Open file GoL-logic-gate-NOT-0-anim.png, ResearchGate, Accessed as on 09-15-2023.

[23] J. Kroc. Logic-gate NOT: inputs 11. https://www.researchgate.net/publication/368300097/, Sep 2023. Open file GoL-logic-gate-NOT-1-anim.png, ResearchGate, Accessed as on 09-15-2023.

[24] J. Kroc. Logic-gate OR: inputs 00. https://www.researchgate.net/publication/367380336/, Sep 2023. Open file GoL-logic-gate-OR-00-anim.png, ResearchGate, Accessed as on 09-15-2023.

[25] J. Kroc. Logic-gate OR: inputs 01. https://www.researchgate.net/publication/367380336/, Sep 2023. Open file GoL-logic-gate-OR-01-anim.png, ResearchGate, Accessed as on 09-15-2023.

[26] J. Kroc. Logic-gate OR: inputs 10. https://www.researchgate.net/publication/367380336/, Sep 2023. Open file GoL-logic-gate-OR-01-anim.png, ResearchGate, Accessed as on 09-15-2023.

[27] J. Kroc. Logic-gate OR: inputs 11. https://www.researchgate.net/publication/367380336/, Sep 2023. Open file GoL-logic-gate-OR-11-anim.png, ResearchGate, Accessed as on 09-15-2023.

[28] J. Kroc. Robust emergents within GoL resisting perturbation of neighborhood. https://www.researchgate.net/publication/372364849/, a 2023. Accessed as on 07-14-2023.

[29] J. Kroc, P.M.A. Sloot, and A. Hoekstra. *Simulating Complex Systems by Cellular Automata*, chapter Introduction to Modeling of Complex Systems Using Cellular Automata, page 406. Understanding Complex Systems. Springer, June 2010. https://link.springer.com/book/10.1007/978-3-642-12203-3.

[30] H.P. Langtangen. *A Primer on Scientific Programming with Python*. texts in Computational Science and Engineering. Springer Berlin, Heidelberg, 2016. Open Access, "https://www.springer.com/book/10.1007/978-3-662-49887-3".

[31] H. Sayama. *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNNY Textbooks, Milne Library, State University of New York at Geneseo, NY, 2015. https://open.umn.edu/opentextbooks/textbooks/233.

[32] J. Sundnes. *Introduction to Scientific Programming with Python*. Simula SpringerBriefs on Computing. Springer Cham, 2020. Open Access, "https://www.springer.com/book/10.1007/978-3-030-50356-7".

[33] T. Toffoli. Cellular automata as an alternative to (rather than and approaximation of) differential equations in modeling physics. *Physica D: Nonlinear Phenomena*, 10(1):117–127, 1984. https://doi.org/10.1016/0167-2789(84)90254-9.

[34] G.Y. Vichniac. Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):96–116, 1984. https://doi.org/10.1016/0167-2789(84)90253-7.